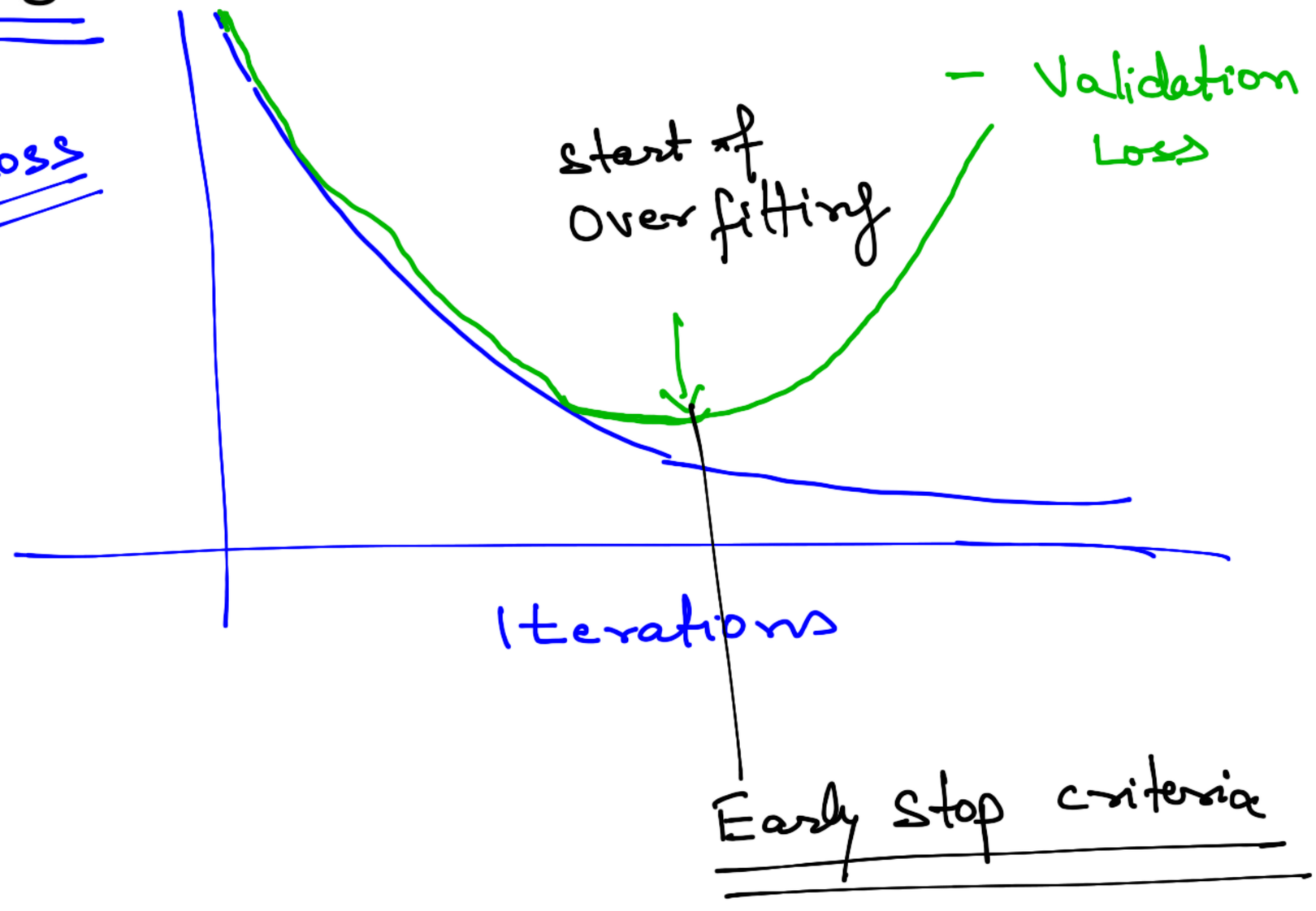
# COL341
# FUNDAMENTALS OF MACHINE LEARNING

Rahul Garg

# Avoiding Overfitting in Neural Networks

## 1 Early stopping



# Avoiding Overfitting in Neural Networks

(2) Regularization

(1) Add $L_2$ penalty

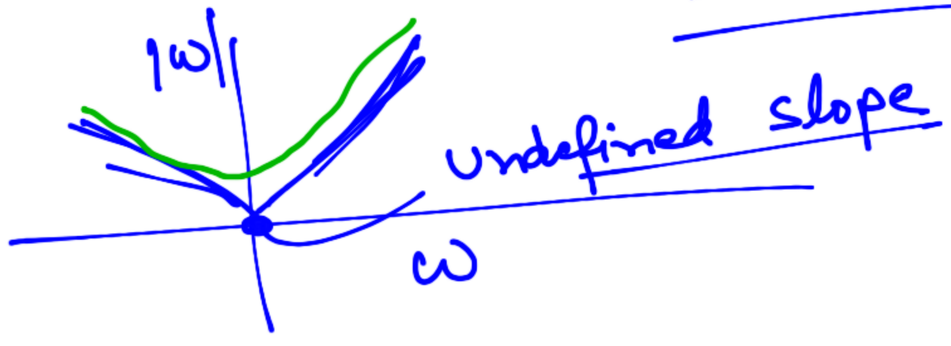
$$= \lambda_2 \sum_j \sum_\gamma \sum_s (w^j_{\gamma s})^2$$

$\lambda_2 > 0$ ← Regularization parameter

(2) Add $L_1$ penalty : Promote Sparsity

$$= \sum_j \sum_\gamma \sum_s \lambda_1 \| w^j_{\gamma s} \|_1$$

Problem: $\frac{\partial}{\partial w}$ is undefined at $w = 0$




(3) Add $L_1 + L_2$ penalty

(4) Add other penalties $\| w^j_{\gamma s} \|^q$ e.g.

# Avoiding Overfitting in Neural Networks

3 Cross validation

- Learning rates
- # units
- # Layers

] Hyper parameters of N.N.

S : Candidate Neural Networks

Make folds.

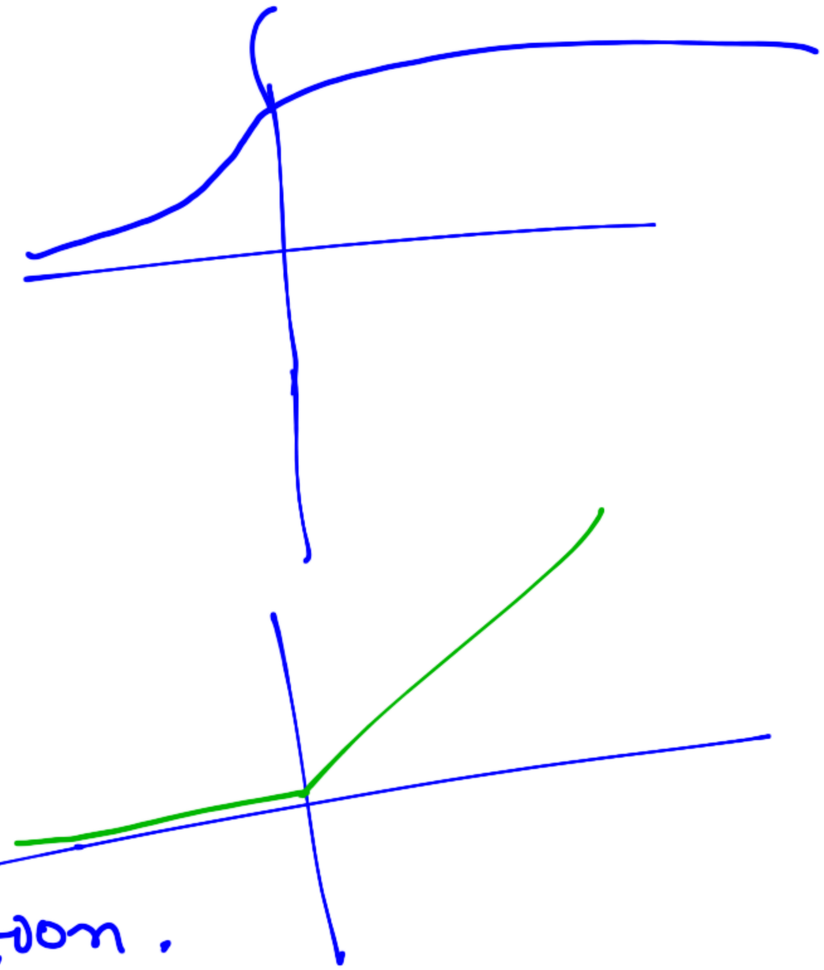
Learn each NN. on each fold. Predict performance on unseen data. ⇒ Early Stopping

(!)

Select best performing NN.

# Activation Functions

1. Sigmoid
2. Tanh
3. RELU



Work out back propagation for your activation function.

# Topics Covered

- Linear Regression
- Ridge Regression
- OLS Solutions
- Gradient Descent
  - Stochastic
  - Mini Batch
  - Batch
  - Choosing step size
    - Fixed
    - Adaptive
      - Exact (closed form)
      - Exact line search
      - $\alpha\beta$ backtracking line search
- Feature Engineering
- Cross Validation

- Classification problems
- Logistic Regression
  - Two Class
  - Multi Class
- Decision Regions / Boundaries
- Convex Sets / Functions
- Testing Convexity
  - PSD
  - Gradients
  - Hessian

Performance Metrics for Classification Problems
- Precision, Recall, sensitivity, Specificity, F Score
- ROC Area, TPR, FPR...

Neural Networks
- Chain rule for vector functions
- Forward & Back propagation
- Learning weights

Exam will be closed book, Closed Internet Invigilated.

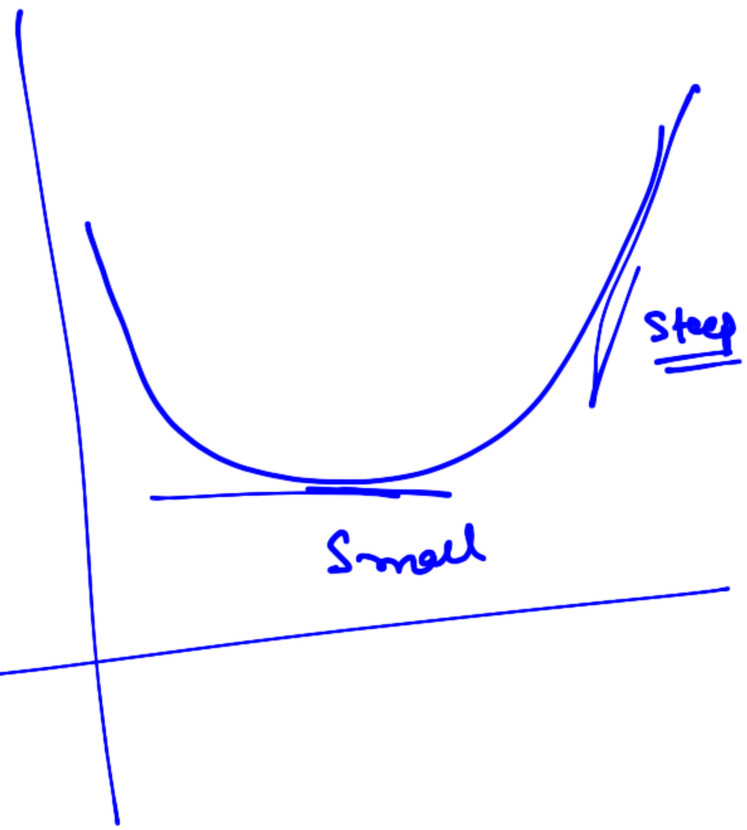
Keep your videos on for the duration of the exam.

MS Teams

# Gradient Normalization

$$g = \frac{\partial L}{\partial w}$$

Normalized $\hat{g} = \frac{g}{\|g\|_2}$

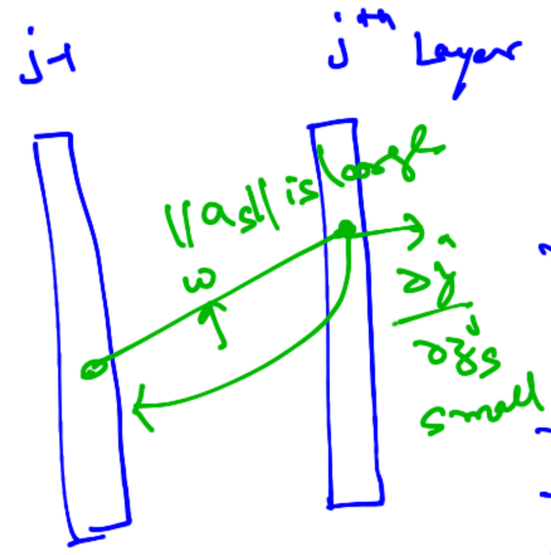



$$w^{t+1} = w^t + \eta_t g^t \quad - (1)$$

$$w^t + \eta_t \hat{g}^t \quad - (2)$$

# Vanishing Gradient Problem

Typical NN's have small # layers

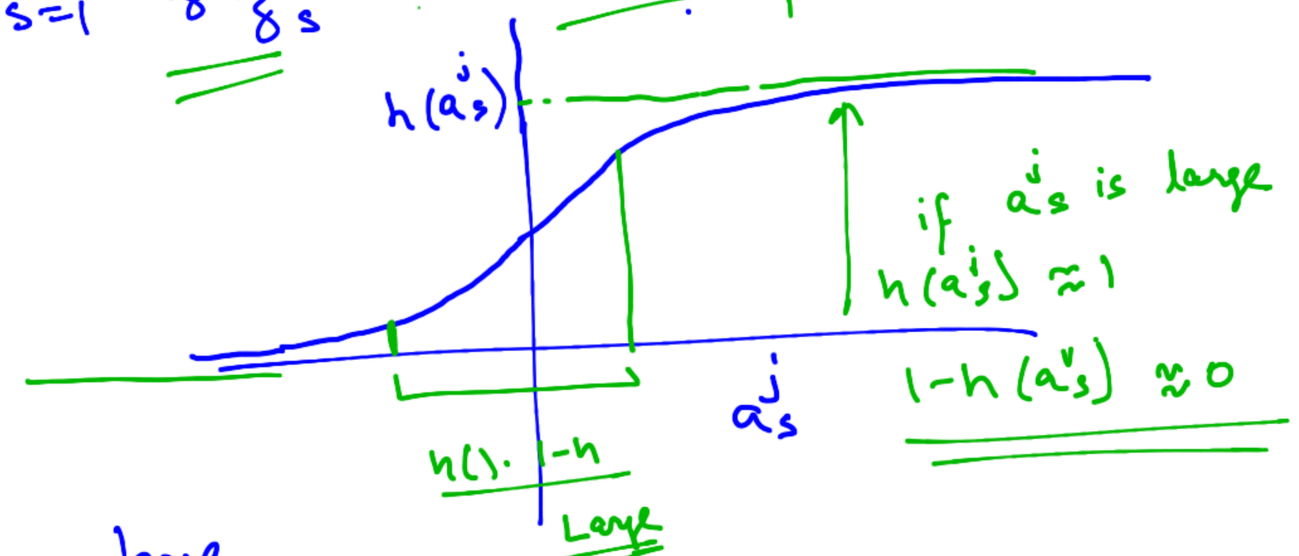



Not contributes to gradient in prev. layer

$$\frac{\partial \hat{y}}{\partial \omega^j_{rs}} = \frac{\partial \hat{y}}{\partial z^j_s} \cdot h(a^j_s)\left[1-h(a^j_s)\right] \cdot z^{j-1}_r$$

$z^{j-1} \cdot \omega^j$

$$\frac{\partial \hat{y}}{\partial z^{j-1}_r} = \sum_{s=1}^{k^j} \frac{\partial \hat{y}}{\partial z^j_s} \cdot h(a^j_s)\left[1-h(a^j_s)\right] \cdot \omega^j_{rs}$$




$\frac{\partial \hat{y}}{\partial \omega^j}$ becomes very small for small $j$'s if # layers is large

HW: ① Read Convolution

1D, 2D, 3D

Discrete Convolution

② Learn Keras / Tensorflow

# Convolutional Neural Networks

Effective in a variety of applications

Image classification, speech and many other domains

Foundations of Deep Learning

Key ideas

- **Convolution layers**
- **RELU activation function**
- **Max-pooling**
- **Deep network**

# Convolutional Neural Networks

General Architecture of CNNs for image problems

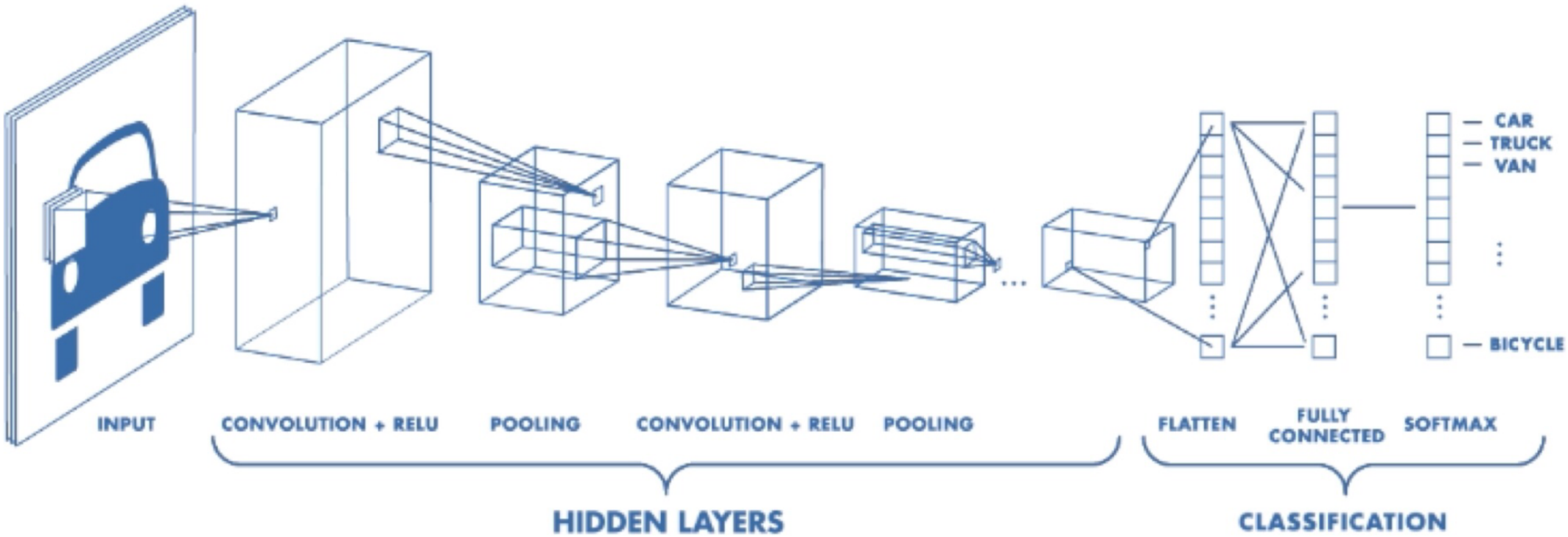
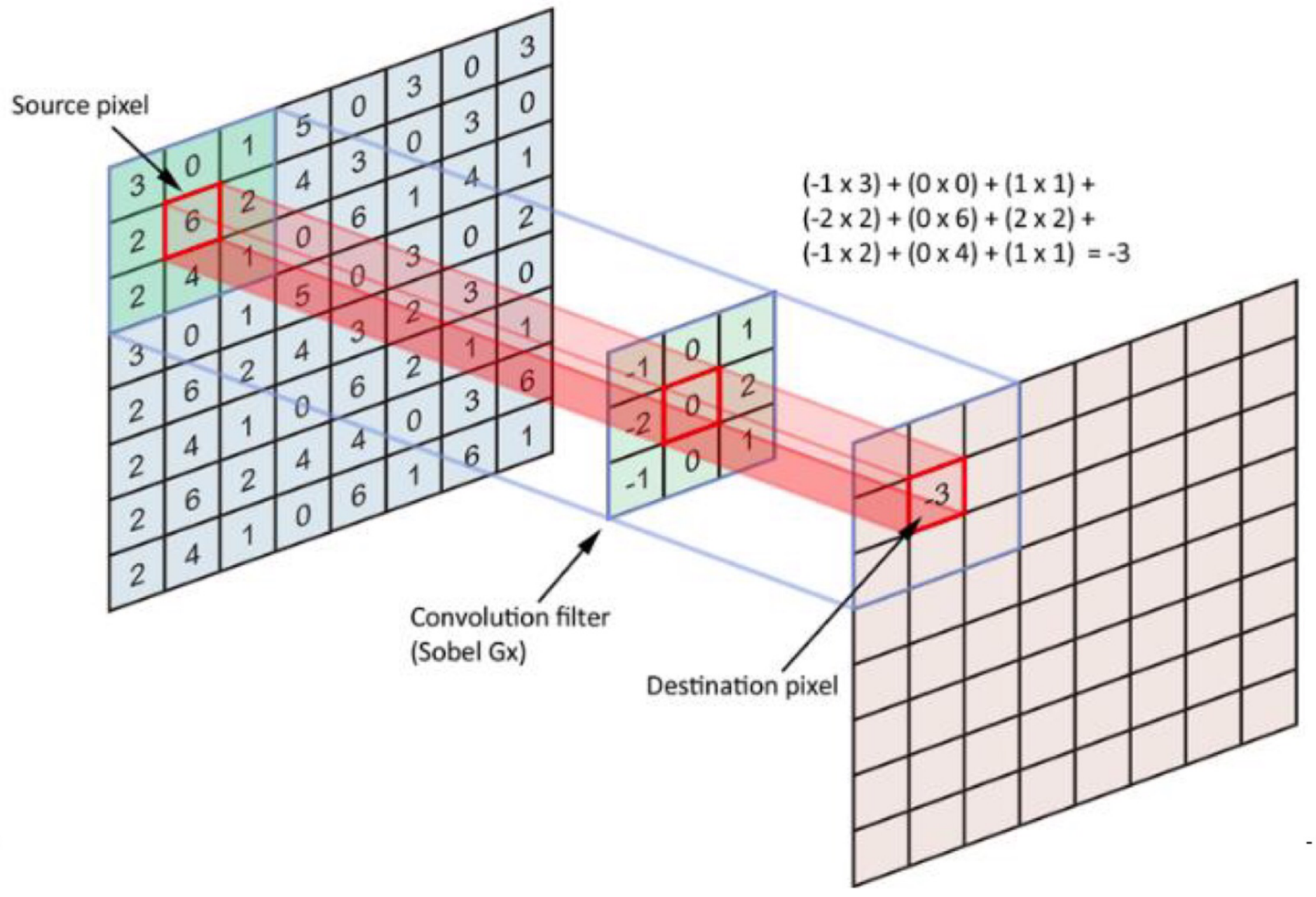



Image source: Mathworks

# Convolution



# 1D Convolution

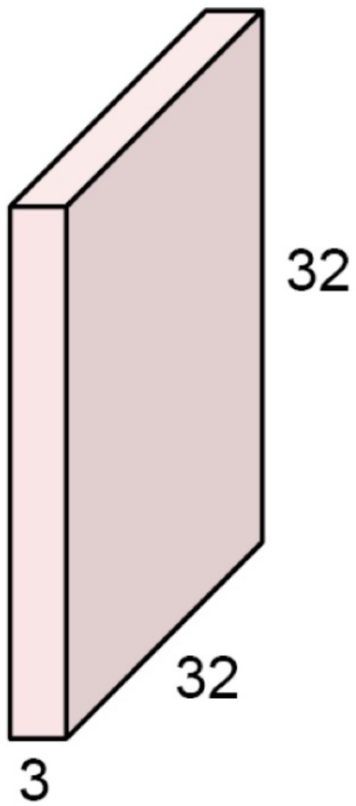
# 2D Convolution

# Shared Weights using Convolution Layer

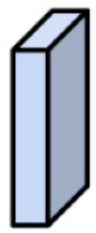
# Convolution Layer

Filters always extend the full depth of the input volume

32x32x3 image



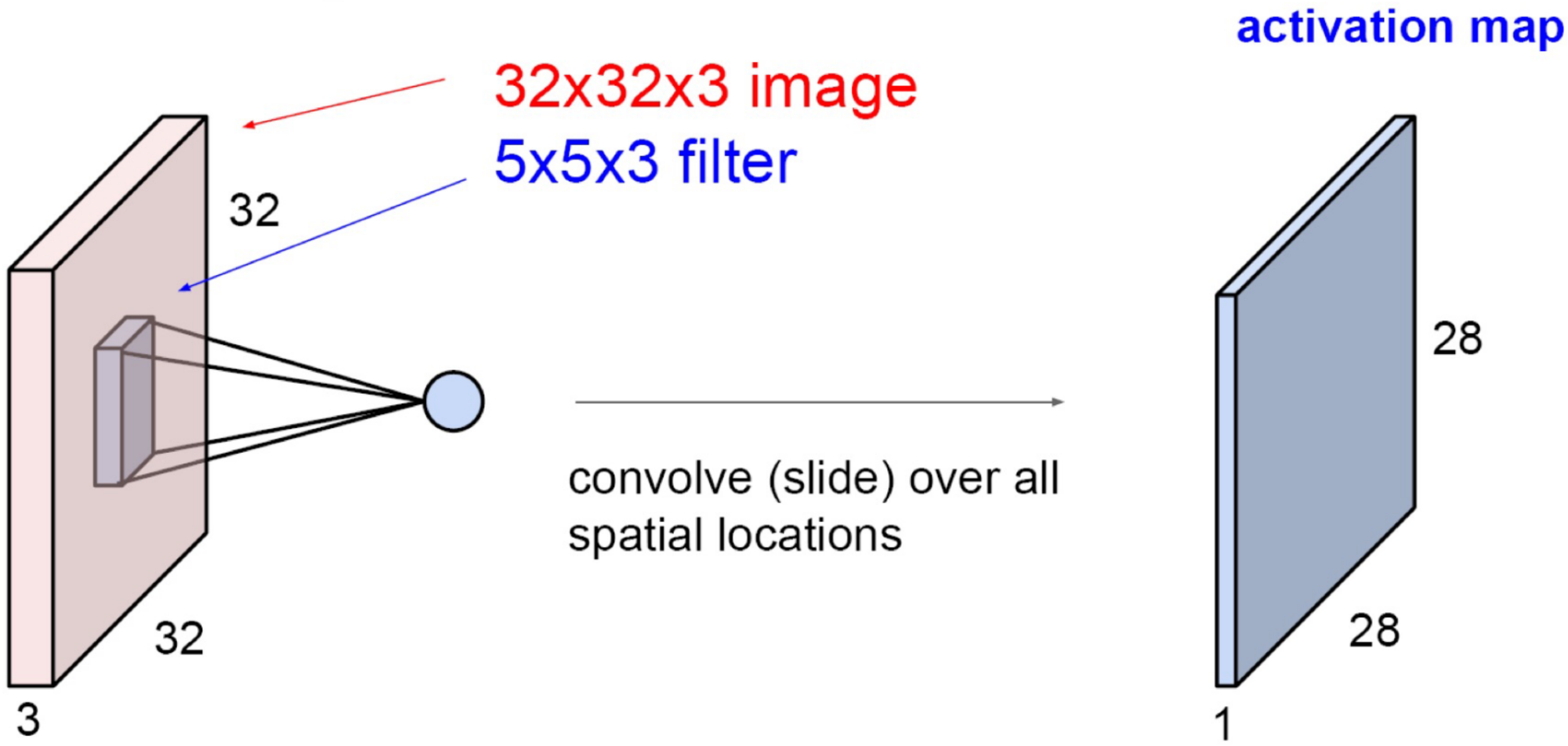

5x5x3 filter



**Convolve** the filter with the image
i.e. “slide over the image spatially, computing dot products”

Adapted from: Fei-Fei Li & Justin Johnson & Serena Yeung, CS231n Stanford University

# Convolution Layer

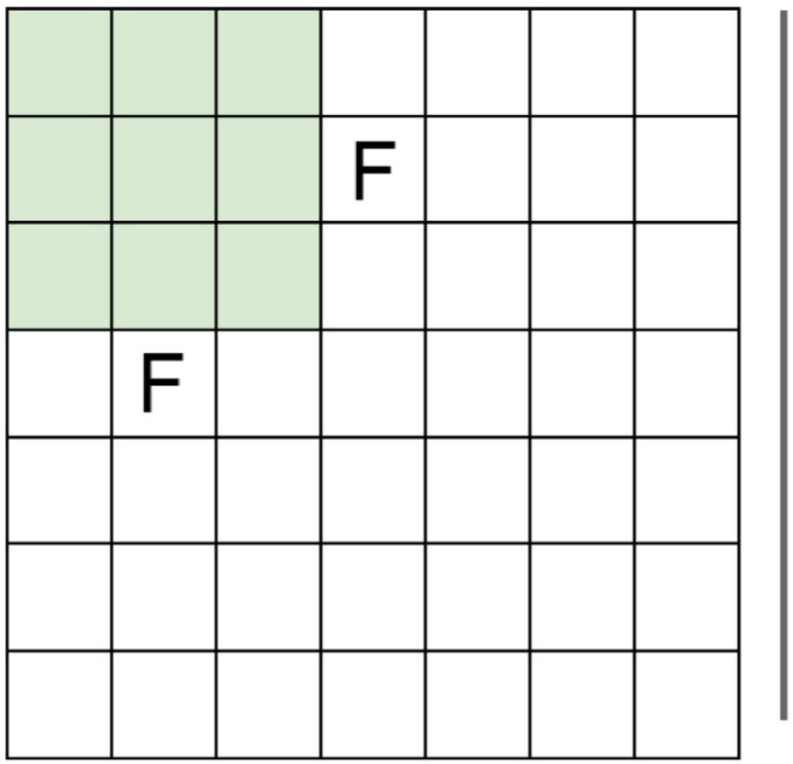



Adapted from: Fei-Fei Li & Justin Johnson & Serena Yeung, CS231n Stanford University

# Convolution Layer

consider a second, green filter


Adapted from: Fei-Fei Li & Justin Johnson & Serena Yeung, CS231n Stanford University

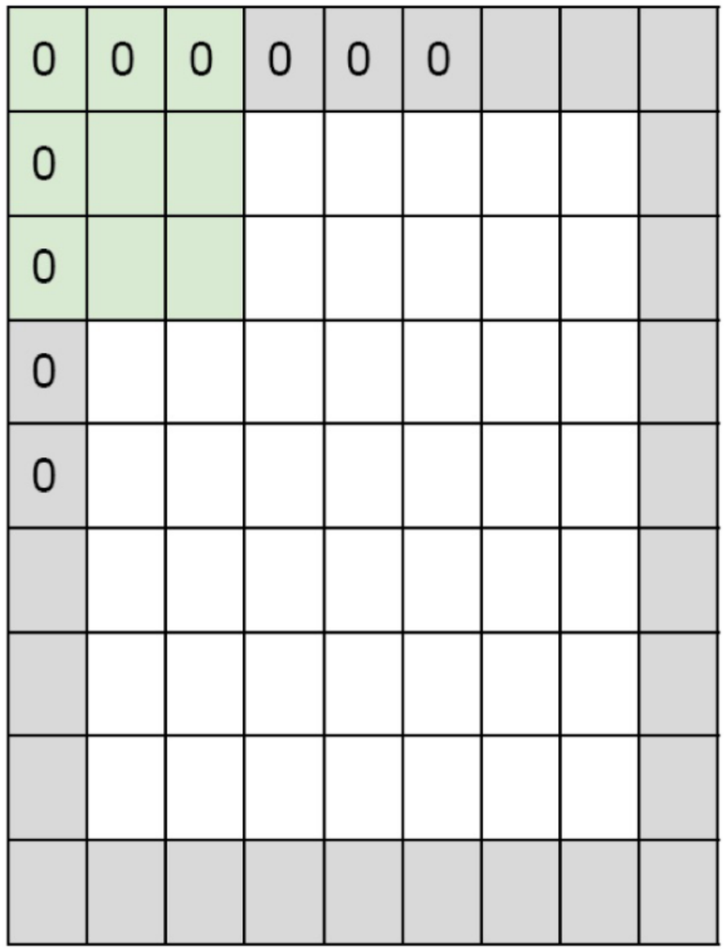



Output size:
**(N - F) / stride + 1**

e.g. N = 7, F = 3:
stride 1 => (7 - 3)/1 + 1 = 5
stride 2 => (7 - 3)/2 + 1 = 3
stride 3 => (7 - 3)/3 + 1 = 2.33

Adapted from: Fei-Fei Li & Justin Johnson & Serena Yeung, CS231n Stanford University

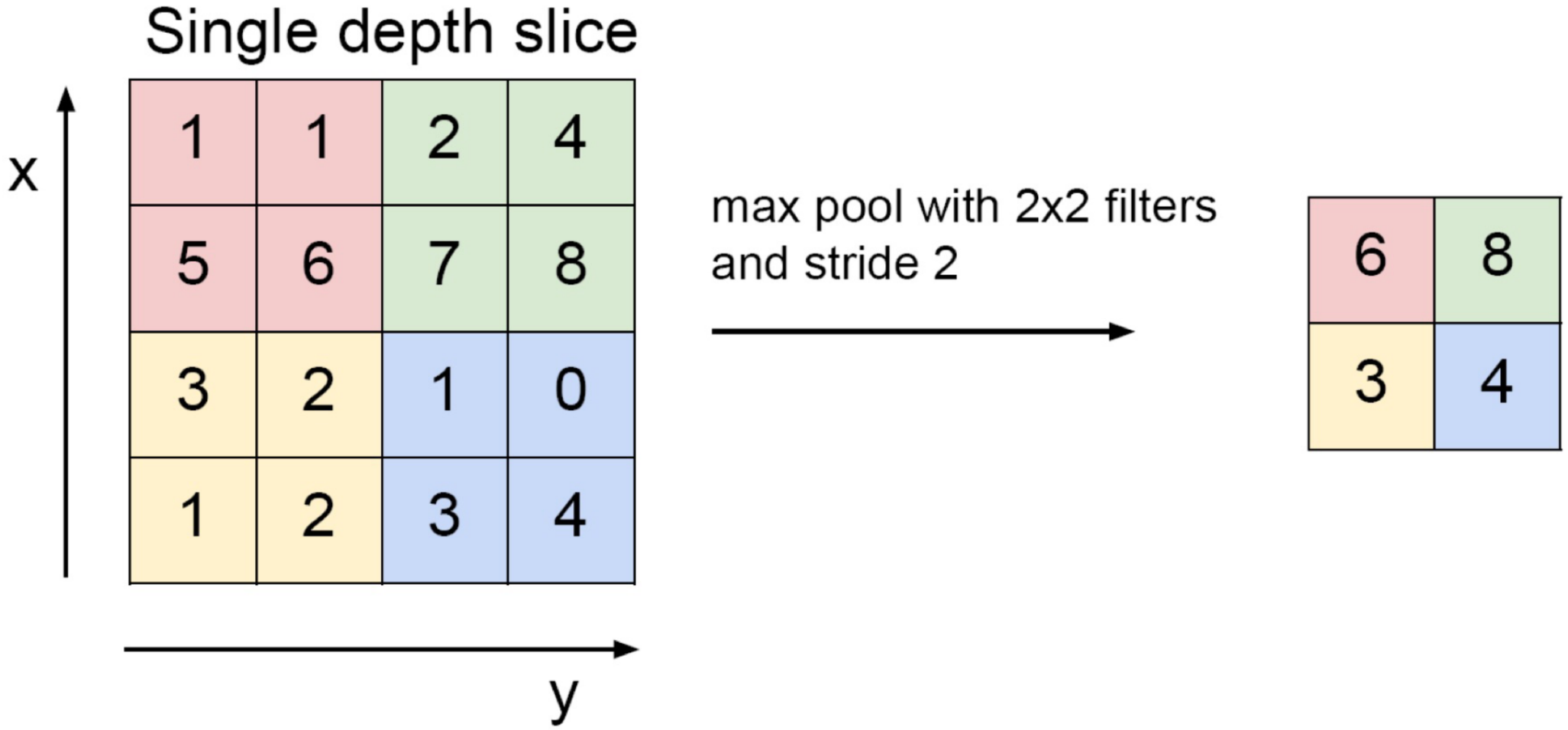
# In practice: Common to zero pad the border




e.g. input 7x7
**3x3** filter, applied with **stride 1**
**pad with 1 pixel** border => what is the output?

**7x7 output!**
in general, common to see CONV layers with stride 1, filters of size FxF, and zero-padding with (F-1)/2. (will preserve size spatially)
e.g. F = 3 => zero pad with 1
F = 5 => zero pad with 2
F = 7 => zero pad with 3

# MAX POOLING




Adapted from: Fei-Fei Li & Justin Johnson & Serena Yeung, CS231n Stanford University

# Max Pooling Parameters

Filter size

Stride

# Backpropagation with Max Pooling

# RELU Variations

$$f(x) = x^+ = \max(0, x),$$ Rectifier linear unit

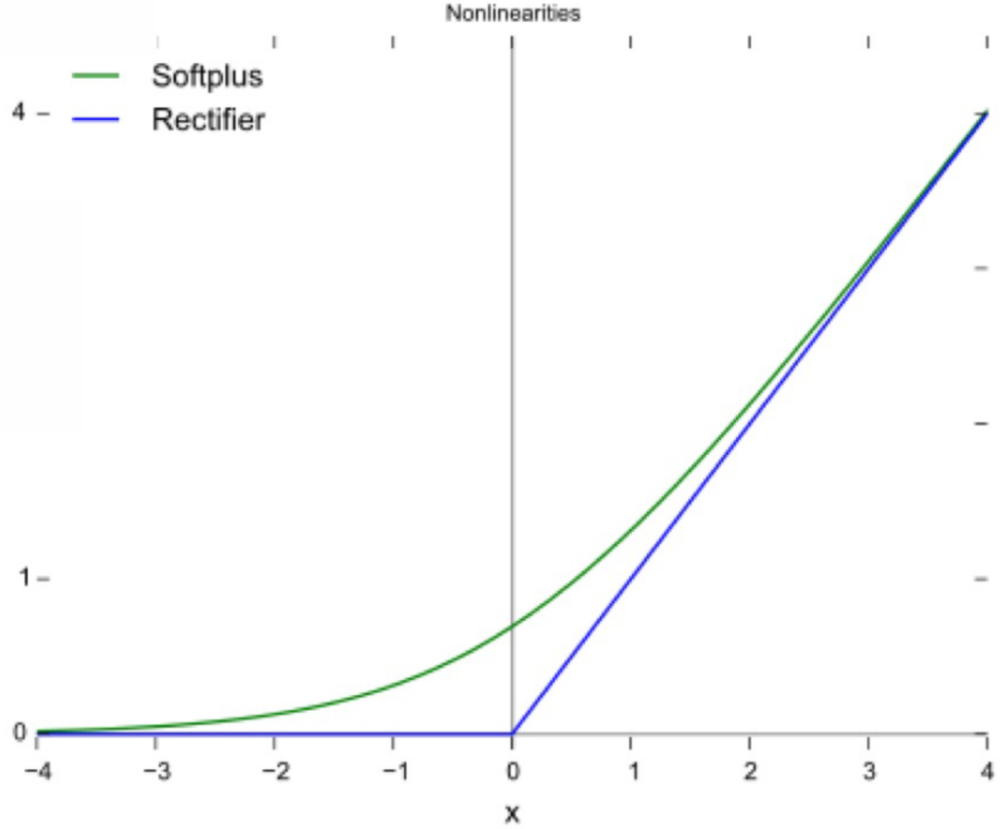
Leads to dead units

$$f(x) = \log(1 + \exp x),$$ Softplus function

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases}$$ Leaky ReLU

Parametric ReLU

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ ax & \text{otherwise} \end{cases}$$



# What happens in CNNs?

**Pre-selected features**

- Frequency domain features
- FFT, DCT etc.
- Wavelet transforms
- Gabor filters
- SIFT features
- HOG features
- Homework: Read about DCT, FFT, Wavelets, Gabor, SIFT, HOG (not in Minor 1)

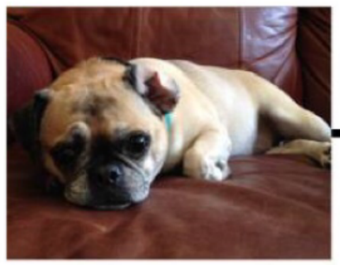
**CNNs**

- The network learns the optimal features for the task
- Successive layers learn more complex features

## Preview

*[Zeiler and Fergus 2013]*

Visualization of VGG-16 by Lane McIntosh. VGG-16 architecture from [Simonyan and Zisserman 2014].

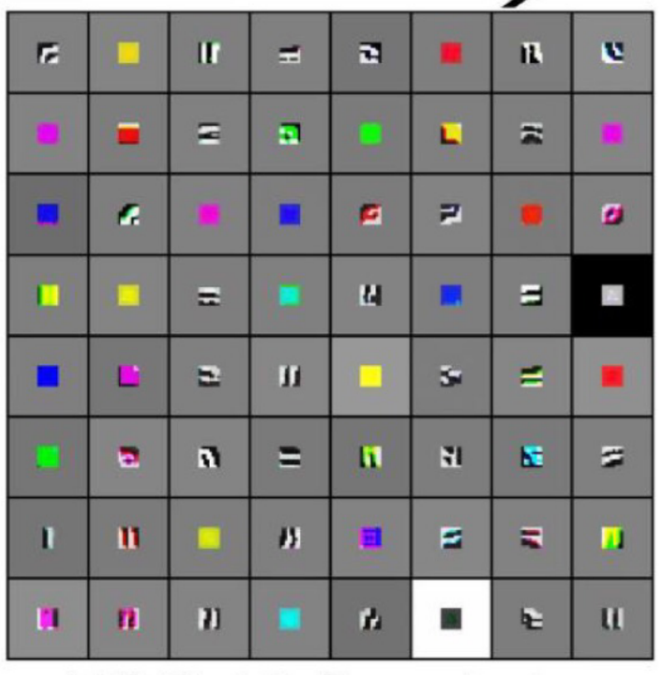


Low-level features

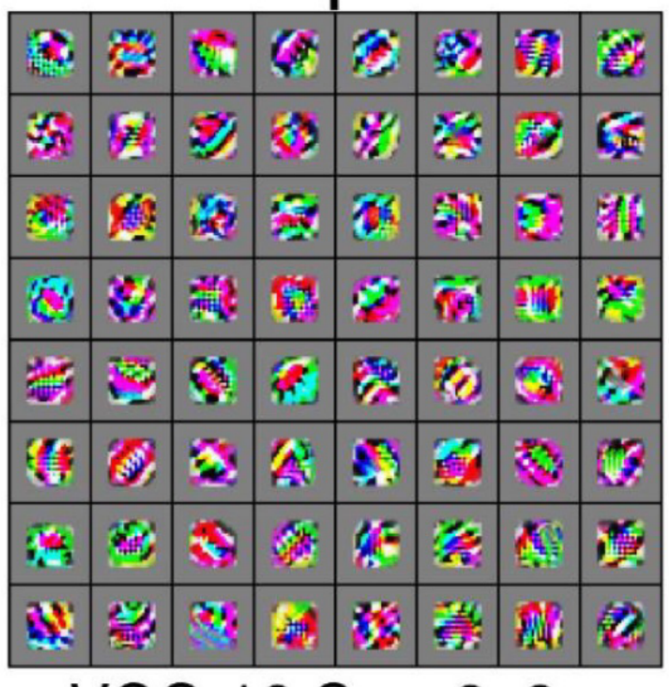
Mid-level features

High-level features

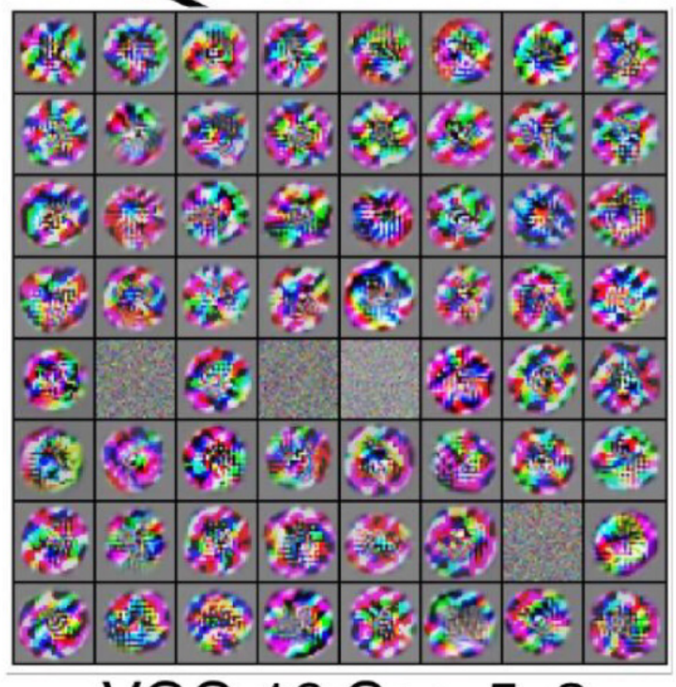
Linearly separable classifier



VGG-16 Conv1_1



VGG-16 Conv3_2

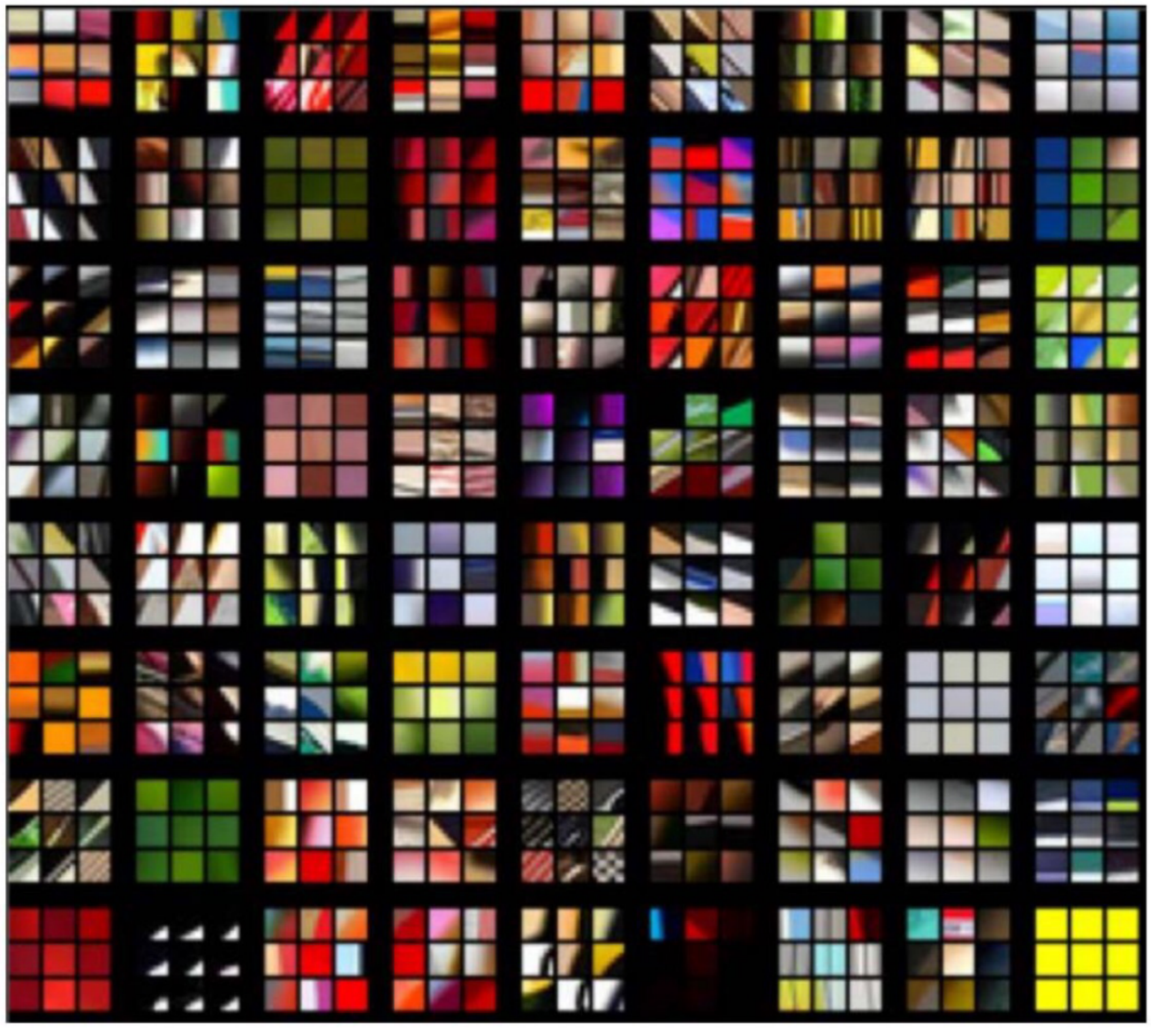


VGG-16 Conv5_3

Adapted from: Fei-Fei Li & Justin Johnson & Serena Yeung, CS231n Stanford University

# Top 9 patches that activate each filter in layer 1

Each 3x3 block shows the top 9 patches for one filter.

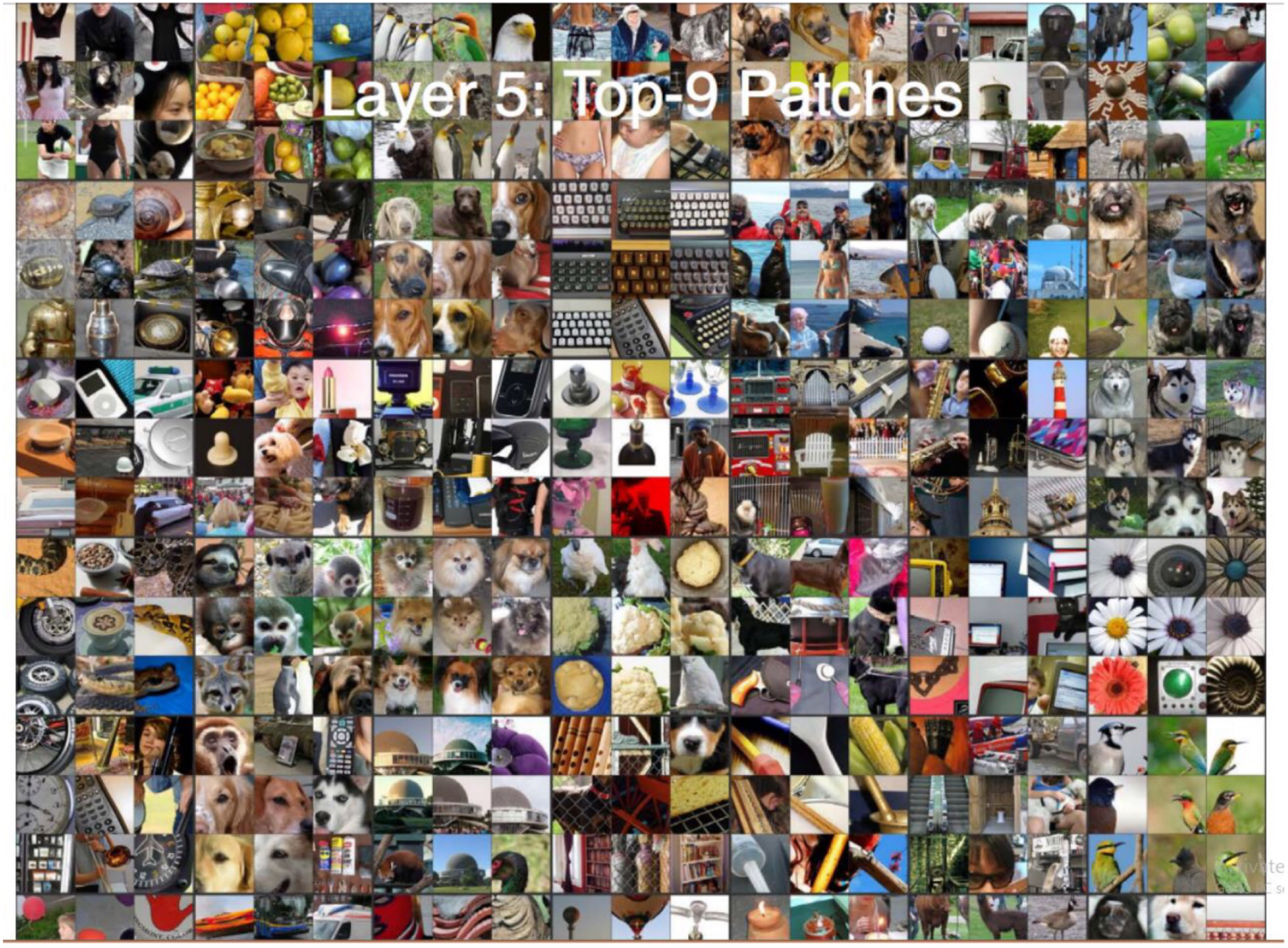




Slide courtesy: David W. Jacobs, University of Maryland


Slide courtesy: David W. Jacobs, University of Maryland


Slide courtesy: David W. Jacobs, University of Maryland




Slide courtesy: David W. Jacobs, University of Maryland